Passed. Good work!

732A38: Computer lab 6
Computational statistics

Caroline Svahn & Martina Sandberg

March 15, 2016

1 Genetic algorithm

In this assignment, you will try to perform one-dimensional maximization with the help of a genetic algorithm.

1.1

Define the function

$$f(x) = \frac{x^2}{e^x} - 2 \cdot exp\left\{\frac{-9sin(x)}{x^2 + x + 1}\right\}.$$

1.2

Define the function "crossover" that for two scalars x and y returns their "kid" as (x+y)/2

1.3

Define the function "mutate" that for scalar x returns the result of the integer division x^2 mod 30. (Operation "mod" is denoted in R as "%%")

1.4

Write a function that depends on the parameters maxiter and mutprob and:

1. Plots function f in the range from 0 to 30. Do you see any maximum value?
2. Defines an initial population for the genetic algorithm as X=(0,5,10,15,. . . ,30)
3. Computes vector "Values" that contains the function values for each population point
4. Performs *maxiter* iterations where at each iteration
   - (a) Two indexes are randomly sampled from the current population, they are further used as parents (use `sample()`.)
   - (b) One index with the smallest objective function is selected from the current population, the point is referred to as victim(use `order()`)
   - (c) Parents are used to produce a new kid by crossover. Mutate this kid with probability *mutprob*. (use `crossover()`, `mutate()`)
   - (d) The victim is replaced by the kid in the population and the list "Values" is updated
   - (e) The current maximal value of the objective function is saved
5. Final observations are added to the current plot and marked by some other color.

1.5

Run your code with different combinations of maxiter=10, 100 and mutprob=0.1, 0.5, 0.9. Observe the initial population and final population. Conclusions?

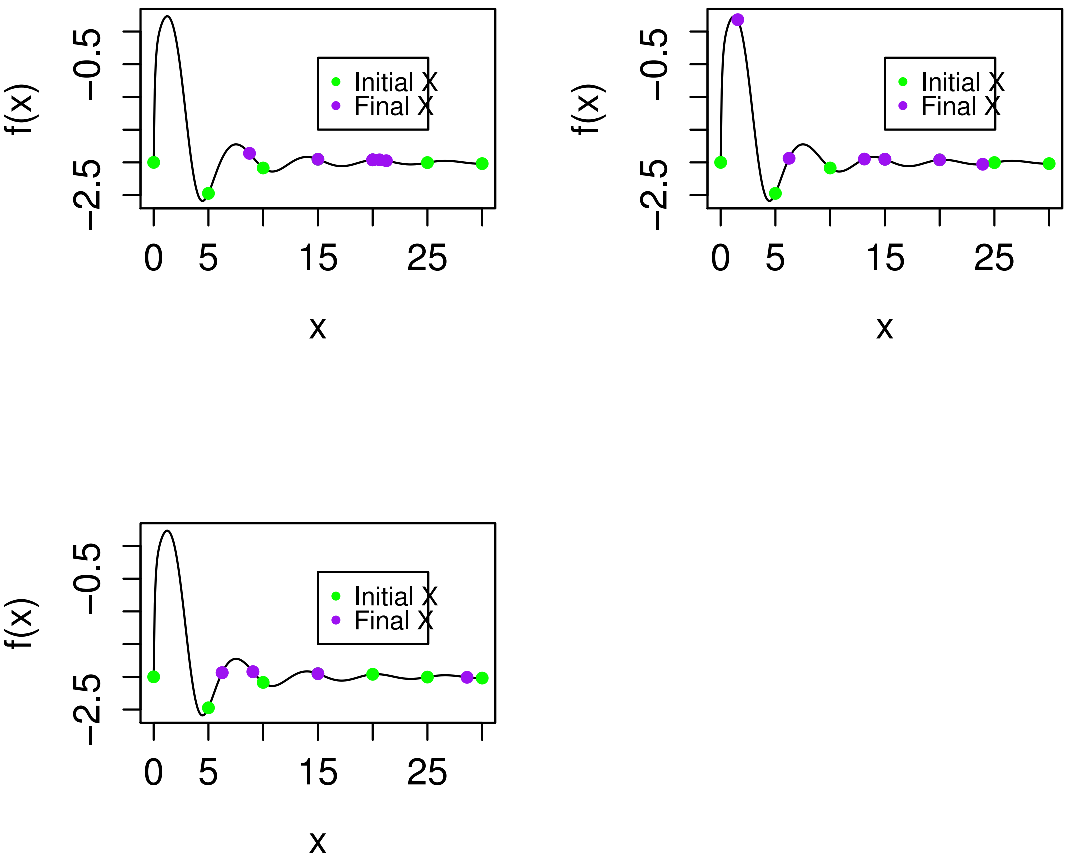

Figure 1: *Output from the implemented function with maxiter=10 and from top left; mutprob=0.1, 0.5, 0.9.*

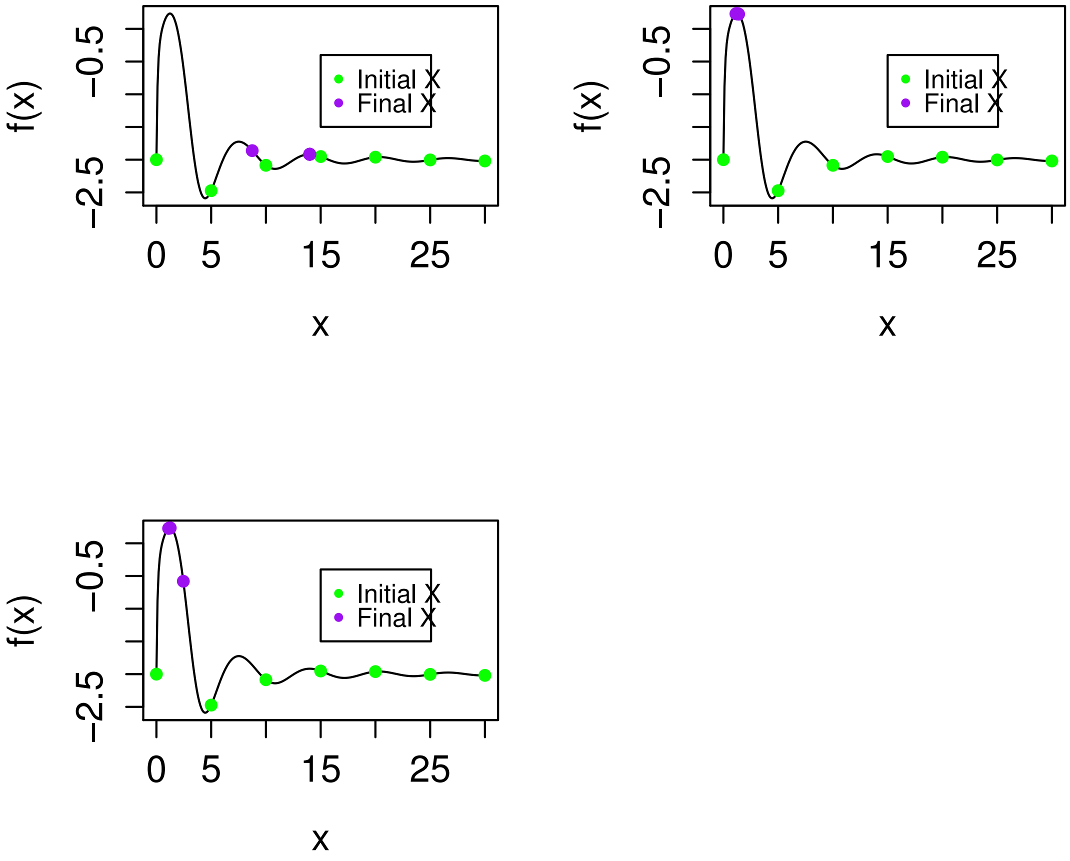

Figure 2: *Output from the implemented function with maxiter=10 and from top left; mutprob=0.1, 0.5, 0.9.*

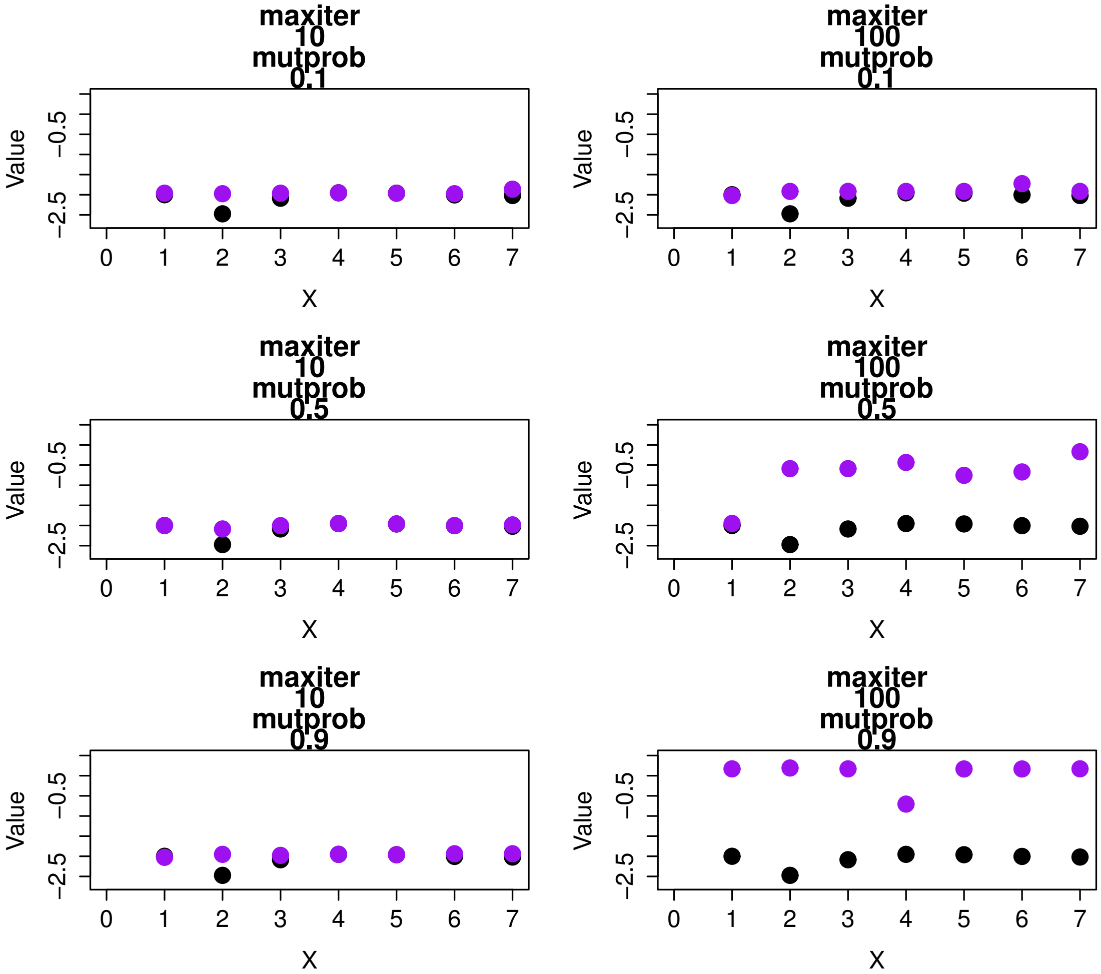

Figure 3: *Value change of the initial population for all combinations of maxiter and mutprob. Black is initial values and purple final values.*

In figure 1 the output from the implemented function with $maxiter = 10$ and $mutprob = 0.1, 0.5, 0.9$ and seed 12345. Here the green dots are the initial population and the purple dots the final population. In figure 2 we can see the same but for $maxiter = 100$. We can conclude that the function $f(x)$ has an obvious max value at approximately 0.23. Of the initial population the maximum value (the best bacteria value) is -1.95.

In figure 3 the output from the implemented function for all combinations of $maxiter$ and $mutprob$ is shown. The black dots are the original bacteria values and the purple the final population. As can be seen, since the population is rather homogeneous, the values stabilize rather quickly. This since the children are the mean of the parents. The number of iterations is important since a mutated kid is needed to create a stronger population, and the probability of a mutation occurring increases with the number of iterations. Increasing the probability of mutation has, naturally, the same effect. As seen in the plot, a high probability of a mutation occurring results in a strong population. With $maxiter = 100$ and $mutprob = 0.9$, the maximum value is 0.19, which is indeed much higher than the maximum value of the original population. After a sufficient number of iterations all final points are going towards the highest point, since this is the only point that never will become the victim.

2 EM algorithm

Data file **physical.csv** describes a behavior of two related physical processes $Y = Y(X)$ and $Z = Z(X)$.

2.1

Make a time series plot describing dependence of Z and Y versus X. Does it seem that two processes are related to each other? What can you say about the variation of the response values with respect to X?

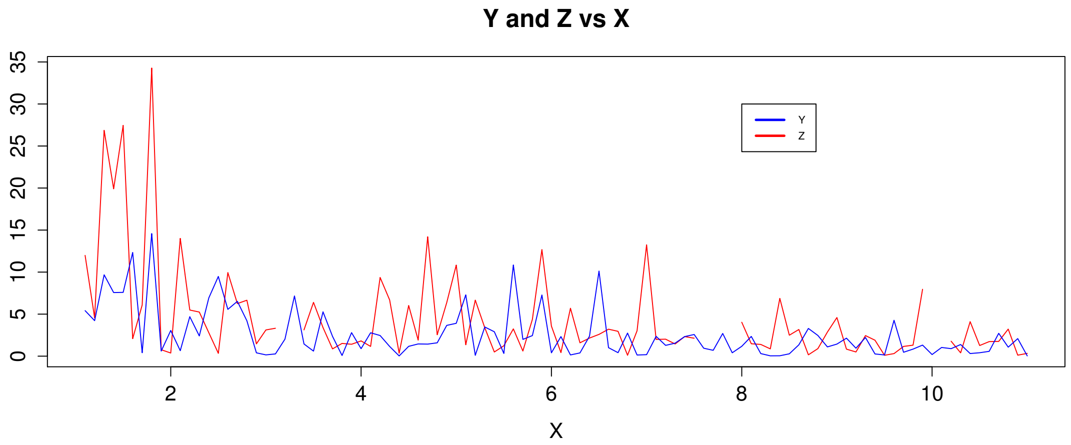

Figure 4: *A time series plot of the physical processes $Y(X)$ and $Z(X)$.*

In figure 4 the two physical processes are plotted. The processes appear to be dependent in some matter since the series follow about the same level, however, the amplitude (variance) appears different. The variance seems to be the same in some parts whereas the obvious differences are in the beginning where Z have a much higher variance than Y and a little higher in other parts. There seems to be some kind of decreasing trend in both of them.

2.2

Note that there are some missing values of Z in the data which implies problems in estimating models by maximum likelihood. Use the following model

$$Y_i \sim Exp\left\{\frac{X_i}{\lambda}\right\}, \quad Z_i \sim Exp\left\{\frac{X_i}{2\lambda}\right\},$$

where λ is unknown parameter, to derive an EM algorithm that estimates λ.

$$\left.\begin{aligned} L(\lambda|Y_i, Z_i) &= \prod_{i=1}^{n} \frac{X_i^2}{2\lambda^2} \cdot exp\left\{\sum_{i=1}^{n} -\frac{X_i}{\lambda}Y_i - \frac{X_i}{2\lambda}Z_i\right\} \\ &= \prod_{i=1}^{n} X_i^2 \cdot \left(\frac{1}{2\lambda^2}\right)^n \cdot exp\left\{\sum_{i=1}^{n} -\frac{X_i}{\lambda}Y_i - \frac{X_i}{2\lambda}Z_i\right\} \end{aligned}\right\} \tag{1}$$

$$\ell(\lambda|Y_i, Z_i) = log(L(\lambda|Y_i, Z_i)) = -2n \cdot log(\lambda) - \sum_{i=1}^{n} \frac{X_i}{\lambda}Y_i - \sum_{i=1}^{n} \frac{X_i}{2\lambda}Z_i + C \tag{2}$$

$$\ell(\lambda|Y_i, Z_i) = -2n \cdot log(\lambda) - \sum_{i=1}^{n} \frac{X_i}{\lambda} Y_i - \sum_{O} \frac{X_i}{2\lambda} Z_i - \sum_{M} \frac{X_i}{2\lambda} Z_i + C \tag{3}$$

$$E_{Z|Y,\lambda_t}(\ell(\lambda|Y_i, Z_i)) = -2n \cdot log(\lambda) - \sum_{i=1}^{n} \frac{X_i}{\lambda} Y_i - \sum_{O} \frac{X_i}{2\lambda} Z_i - |M| \frac{\lambda_t}{\lambda} + C \tag{4}$$

$$\frac{\delta E}{\delta \lambda} = -\frac{2n}{\lambda} + \sum_{i=1}^{n} \frac{X_i}{\lambda^2} Y_i + \sum_{O} \frac{X_i}{2\lambda^2} Z_i + |M| \frac{\lambda_t}{\lambda^2} \tag{5}$$

$$\lambda = \frac{1}{2n} \left(\sum_{i=1}^{n} X_i Y_i + |M|\lambda_t + \frac{1}{2} \sum_{O} X_i Y_i \right) \tag{6}$$

The likelihood for these values can be written as (1). To get the log likelihood we take log of this and get (2), where C is a constant that contains values that do not depend on λ. Since Z has some missing values, we need to separate the sum containing z_i into two sums, where one contains the observed values (O) and one the missing values (M), which can be seen in (3). Of course, we do not know the values of the missing z_i, so we substitute this with the expected sum of the missing values, which we can be compute as $E(\sum_M \frac{X_i}{2\lambda} Z_i) = \sum_M \frac{X_i}{2\lambda} E(Z_i) = \sum_M \frac{X_i}{2\lambda} \cdot \frac{2\lambda_t}{X_i} = \sum_M \frac{\lambda_t}{\lambda} = |M| \frac{\lambda_t}{\lambda}$, where $|M|$ is the number of missing values. So we have (4). To optimize, we take the derivative of this equation with respect to λ and get (5). Finally, to get the maximal value of λ we put the expression to zero and solve for λ. The obtained λ-value is (6).

2.3

Implement this algorithm in R and use $\lambda_0 = 100$ and convergence criterion "stop if the change in λ is less than 0.001". What is the optimal λ and how many iterations were required to compute it?

In the algorithm we start with an initial λ_t. With this the function (6) is used to compute λ. If the difference between this λ and the initial λ_t is smaller than 0.001 we will stop and conclude that we have found an optimal λ, if its not we will continue and set the computed λ to λ_t. When we implement this we get that the optimal λ is approximately 10.696, and it is obtained after 4 iterations.

2.4

Plot EY and EZ versus X in the same plot as Y and Z versus X and comment whether the computed λ seems to be reasonable.

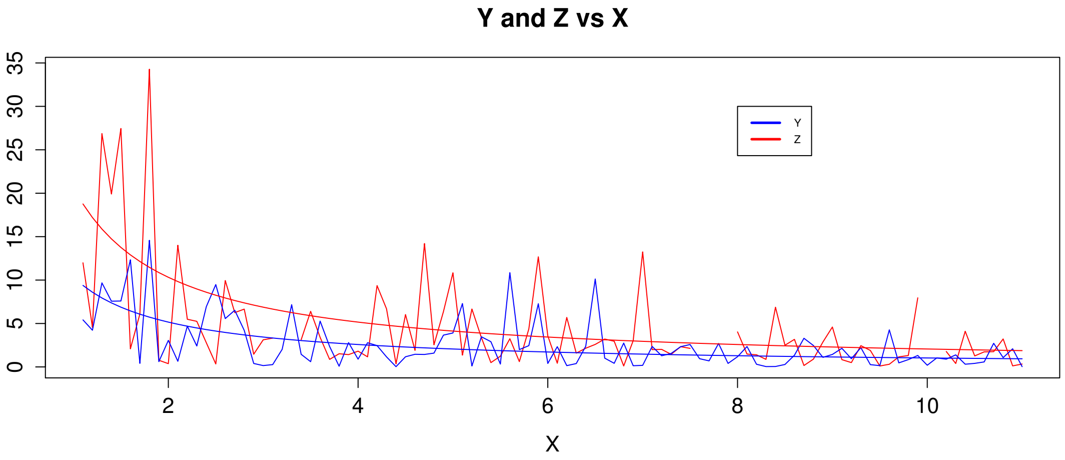

Figure 5: *A time series plot of the physical processes $Y(X)$ and $Z(X)$.*

In figure 5 the two physical processes are plotted again but here with each respective mean value computed with the optimal λ. It seems like this values is reasonable, since they follow the trends in the data. Regarding the distributions used for Y and Z - visually, one could see that the distributions appeared to be similar but that the amplitudes ought to act differently. The distribution used for Z included twice the value of λ whilst Y only used λ - this is then reasonable.

Appendix

Contribution

Most of the plots are from Martina's report but some are from Caroline's. The descriptions of the algorithms, equations and interpretations are merged from both reports. The code is Caroline's but Martina's code is similar and yields the same results. Only one students code is used so that the results should be consistent for the tasks - especially since the code is quite trivial, and the possible variations are few.

Code

```
#1.1

fx<-function(x) {
  y<-(x^2/exp(x))-2*exp((-9*sin(x))/(x^2+x+1))
  return(y)
}

#1.2

crossover<-function(x,y) {
  kid<-(x+y)/2
  return(kid)
}

#1.3

mutate<-function(x) {
  mutated<-x^2%%30
  return(mutated)
}

#1.4

myBacteries<-function(maxiter,mutprob) {
  x<-seq(0,30,5)
  func<-(x^2/exp(x))-2*exp((-9*sin(x))/(x^2+x+1))
  plot(func, xlim=c(0,7), ylab="Value", xlab="X", ylim=c(-2.7,0.5),
              pch=19, cex=2, main=c("maxiter",maxiter,"mutprob",mutprob))
  values<-fx(x)
  max<-numeric(maxiter)

  for (i in 1:maxiter) {
    indices<-1:length(values)
    parents<-sample(indices,2)
    victim<-order(values)[1]

    kid<-crossover(x[parents[1]], x[parents[2]])
    U<-runif(1)

    if (mutprob>=U) {
       kid<-mutate(kid)
    }

    x[victim]<-kid
    values<-fx(x)
    max[i]<-max(values)
  }
  points(values, col="purple", pch=19, cex=2)
  return(c(max(max), min(values)))
}

par(mfrow=c(3,2))
myBacteries(maxiter=10, mutprob=0.1)
myBacteries(maxiter=100, mutprob=0.1)

myBacteries(maxiter=10, mutprob=0.5)
myBacteries(maxiter=100, mutprob=0.5)

myBacteries(maxiter=10, mutprob=0.9)
myBacteries(maxiter=100, mutprob=0.9)

#2.1
phys<-read.csv("physical.csv", sep=",")

par(mfrow=c(1,1))

plot(phys$X,phys$Z, type="l", col="seagreen",main="Y␣and␣Z␣versus␣X",
        ylab="", xlab="X")
lines(phys$X,phys$Y, col="purple")
legend(x=10.02,y=35.7,legend=c("Z", "Y"), lty=c(1,1),
        col=c("seagreen", "purple"))


#2.3
sY<-sum(phys$X*phys$Y)
sZ<-sum(phys$X*phys$Z, na.rm=TRUE)
n<-nrow(phys)
```

```
lambda<-numeric()
lambda[1]<-100
t<-2
M<-sum(is.na(phys$Z))

repeat {
lambda[t]<-(sY+0.5*(sZ+lambda[t-1]*M))/(2*n)

  if (abs(lambda[t-1]-lambda[t])<0.001) {
    break()
  }
  t<-t+1
}

#2.4
EY<-(lambda[t])/phys$X
EZ<-(2*lambda[t])/phys$X

plot(phys$X,phys$Z, type="l", col="seagreen",
        main="Expected␣values␣of␣Y␣and␣Z", ylab="", xlab="X")
lines(phys$X,phys$Y, col="purple")
lines(phys$X,EZ, col="seagreen", lty=2)
lines(phys$X,EY, col="purple", lty=2)
legend(x=9.8,y=35.7,legend=c("Z", "Y", "EZ", "EY"), lty=c(1,1,2,2),
        col=c("seagreen", "purple"))
```